Hierarchical Power Optimization for System-on-a-Chip (SoC) through CMOS Technology Scaling*

Kyu-won Choi and Abhijit Chatterjee
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
{kwchoi,chat}@ece.gatech.edu

Abstract

This report describes an efficient hierarchical design and optimization approach for ultra-low power and minimum area CMOS logic circuits in a system-on-a-chip (SoC) design environment. For state of the art systems, the trade-off solutions between the conflicting design criteria (Delay, Area, and Power) should be considered. In this report, we consider interactions between abstraction levels of the design hierarchy and present techniques that co-optimize the power and the area without performance degradation through judiciously explored technology parameters: Supply voltage, Threshold voltage, and Device width. Experimental results deliver over an order of magnitude savings in power over conventional optimization methods.

I. Introduction

System-on-a-chip is the main technology theme of the semiconductor industry for providing multimedia and communication products for the twenty first century. CMOS technology as a platform for SoC is now required to have a wide range of performances in support of high-speed, minimum-space, and low-power operations [1,2]. We propose a hierarchical design strategy (from RTL level to device level) for low-power SoCs throughout this report.

Exploration of the interaction between device technology and power/area aware electronics is a relatively recent branch of SoC design automation research [3-5]. For device engineers, this research may contain lessons for how to optimize the technology. For circuit designers, a more accurate understanding of device performance limitations and new possibilities both for the present and the future should emerge. In this

* This work is supported in part by DARPA (Defense Advanced Research Projects Agency) under the grant #E21-F48

report, we propose a power/area co-optimization scheme at the circuit level through the device level technology parameters scaling: supply voltage (Vdd), threshold voltage (Vth), and device width (W).

Traditionally, the computation of the entire area/power versus delay trade-off of the circuit critical paths has been avoided because of its dynamic nature, i.e., the critical path changes with the optimization, false path variations with the input vectors, and the exponentially increasing path numbers with the gate sizes. For efficient power/area reduction, there are a number of heuristic or combined algorithmic/heuristic optimizers at each abstraction design level [4,5]. In this report, we demonstrate a hierarchical approach to solve the trade-off problems, especially for low-power optimization. Fig. 1 shows the hierarchical "Feed Forward" design approach and methodology that we followed on this report for ultra-low power and design efficiency.

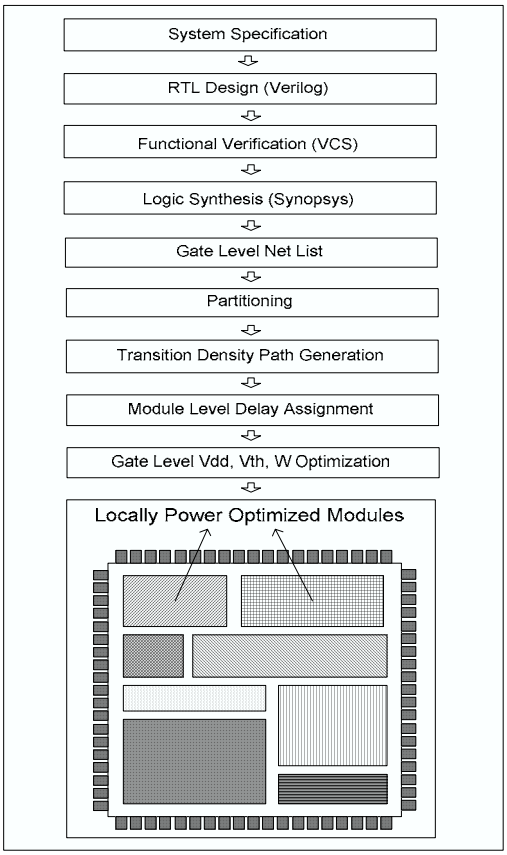


Figure 1. Hierarchical Design Flow for Ultra-low Power

We propose a novel approach for minimizing the total of the static, the dynamic, and the short-circuit power dissipation components in a CMOS logic network required to operate at a specified clock frequency. Fig. 2 shows the overall hierarchical optimization approach. First of all, for the low-power optimization procedure, hierarchical delay assignment is critical because the power reduction is determined by the assigned maximum delay for each module and minimization of the slack time for the each module at the hierarchical design flow. The slack time means the difference between the signal required time and the signal arrival time at the primary output of each module. We will explain the optimization rationale in the later section (Section III) more closely. We introduce the Transition Density Path (TDP) based delay assignment scheme for each module for the best power reduction because dynamic power consumption depends mostly on the switching activities. We compare the performance with two conventional approaches for the delay assignment: 1) critical delay based scheme and 2) fan-in/out counts based scheme.

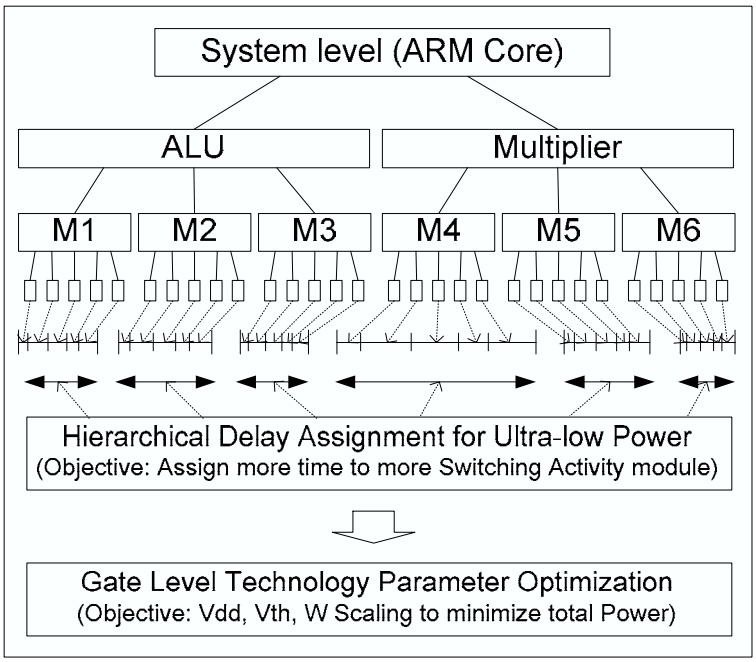


Figure 2. Hierarchical Optimization Approach

For the gate-level optimization step in Fig. 2, a methodology for minimizing the sum of static or leakage and dynamic energy consumption without regard to a performance requirement are proposed. Total power is minimized through selection of supply and threshold voltage values such that the leakage and switching components of the dissipation are equal. The accompanying performance loss can be overcome to some degree by minimizing the product of the switching energy and the propagation delay instead of power or energy alone. The relationship between transistor sizing and power is also examined. In this report we describe a strategy for solving the following problem for the gate level optimizer:

- Given: 1) a random logic network of N static CMOS gates, 2) a required operational clock frequency f 3) a device technology and 4) activity profiles at each input node.
- Determine: 1) the supply voltage Vdd, 2) the threshold voltage Vth of each MOSFET, and 3) the channel width W of each MOSFET such that the sum of the static, dynamic and short-circuit components of energy consumption in a clock cycle is minimized while allowing operation at the desired clock frequency f .

The number of distinct threshold voltages that are allowed by the tolerable technology complexity is also specified. For simplicity of fabrication and design, it is desirable that all the gates in the logic network have identical threshold voltage. Increasing the number of distinct threshold voltages incurs proportional escalation of processing or design complexity, requiring, for example, additional implant masking steps, generation and application of multiple tub biases [6], or migration to a triple-tub process.

The resulting designs operate at low supply voltages and have comparable leakage and switching power dissipation components. The leakage current becomes significant due to the need to reduce threshold voltage with reduced supply voltage to maintain speed. The proposed optimization algorithms and the associated CAD tools allow an order of magnitude reduction in power consumption over designs optimized for only supply voltage and device widths (as opposed to supply voltage, device widths and threshold voltage).

This report is organized as follows including the sub-sections: Theoretical background and previous work; Motivation and its problem solution; Key contribution; Methodology and algorithm; Experimental results and discussion; and finally, Conclusion and future work.

II. Theoretical Background and Prior Work

1. Physics of the Technology Scaling

To illustrate the dynamics of the power minimization process, let us consider a fully loaded static CMOS gate resident in a random logic network and required to operate at a specified clock rate frequency. The desired clock frequency constrains the delay of the gate to not exceed a certain value. For the purpose of illustration, the activity factor of the gate is assumed to be known. Lowering the supply voltage causes the dynamic component of the dissipation to reduce quadratically. However, at very low values of the supply voltage, the threshold voltage must be reduced considerably, causing the leakage dissipation to increase exponentially. In addition, an increase in device width contributes to larger static dissipation and to some extent prevents the dynamic power component from reducing quadratically. Therefore, the sum total of the static and the dynamic components of dissipation is minimized by a unique choice of supply voltage, threshold voltage and device width values. At this optimum configuration, the sum of the increased static dissipation due to lower threshold voltage and the increased static and dynamic dissipation due to larger device width equals the reduction in the dynamic power due to power supply voltage scaling.

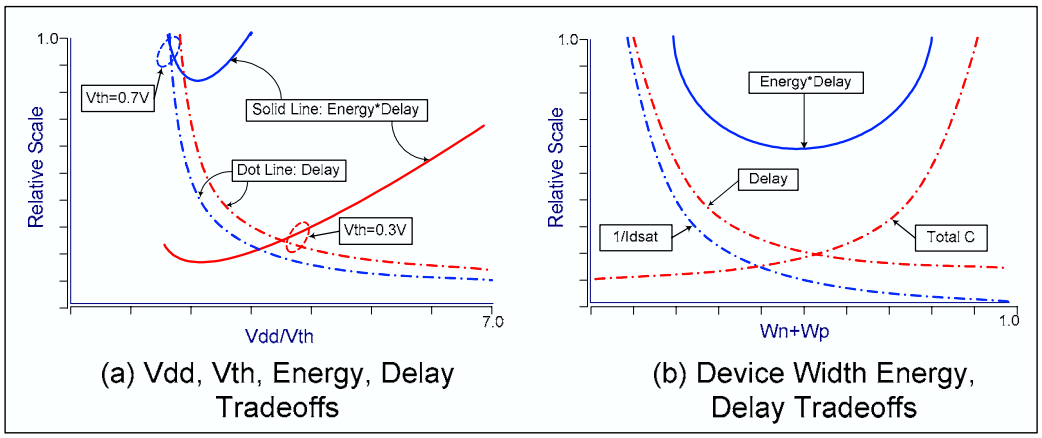


Figure 3. Technology Parameters Tradeoffs [7]

Fig. 3 presents the fundamental characteristics of those three device parameters for power and delay tradeoff. Fig 3(a) shows the Vdd/Vth and Delay*Energy tradeoffs and Fig 3(b) shows the Device Width and Delay*Energy tradeoffs. In this report, we try to optimize the non-linearity of those tradeoffs efficiently to minimize the total power.

2. Supply Voltage Scaling and its Tradeoff / Limitation

Supply voltage scaling technique for low power has been investigated in almost all levels of the design hierarchy from system level to device level due to the quadratic effect on the switching power component. Many respective researches have been shown up in literature [8]. However, it does not come without penalties [9]. The scaling limitations of Vdd reduction are: 1) Delay increase (performance requirements impose a limit); and 2) Noise margins decrease (circuit is more susceptible to noise related soft failures). The approaches to overcome the extent of Vdd scaling are: 1) Availability of high-efficiency DC-DC converter for use [10]; 2) Scaling down the dimensions of devises along with Vdd to compensate for the effects of Vdd on performance; and 3) Reduction of the threshold voltage of transistors.

3. Threshold Voltage Scaling and its Tradeoff / Limitation

Threshold voltage scaling can be used to compensate the performance penalty of the Vdd reduction. In addition, for the active mode of operation, the low Vth is preferred because of the higher performance. However, for the standby mode, high Vth is useful for reduction of leakage power. Different threshold voltages can be developed by multiple Vth implantation during the fabrication, by changing the substrate and source bias, by controlling the back gate of double-gate SOI (silicon on insulator) devices [10]. Some techniques in literature are: 1) SATS (self adjusting threshold voltage scheme) [11]; 2) MTCMOS (multi-threshold voltage CMOS) [12]; 3) DTMOS (dynamic threshold voltage MOSFET) [13]; and 4) DGDT-SOI (double fate dynamic threshold control SOI) [14]. In general, the threshold voltage is a function of a number of parameters including the following: 1) Gate conductor, 2) Gate insulation material, 3) Gate insulator thickness-channel doping, 4) Impurities at the silicon-insulator interface, and 5) Voltage between the source and the substrate.

4. Device Width Scaling and its Tradeoff / Limitation

Transistor and gate sizing affects for dynamic and leakage power reduction and delay. A large gate is required to drive a large load capacitance with acceptable delay but requires more power. The basic rule is to use the smallest transistors or gates that satisfy the delay constraints [15]. To reduce dynamic power, the gates that toggle with higher frequency should be made smaller. An interesting problem occurs when the sizing goal is to leakage power of a circuit. The leakage current of a transistor increases with decreasing threshold voltage and channel length. In general, a lower threshold or shorter channel transistor can provide more saturation current and thus offers a faster transistor. This presents a tradeoff between leakage power and delay. There have been a number of optimization algorithms for gate sizing for dozens of years [16].

It is clear that for any optimization tool to produce coherent results, it must be able to accurately model the effects of the various parameters on the delay of a CMOS gate and its total power dissipation. In this report we have used accurate models for static, dynamic and short-circuit dissipation components, as well as the subthreshold and superthreshold delay of CMOS gates described in [17]. The models that we used for calculating the short-circuit power dissipation were adapted from [18]. All models have been verified by comparison with Hspice [17]. The models are described in detail in the appendix.

III. Motivation and Problem Solution

The objective of this report is to present a technique that minimizes the sum of the static, dynamic and short-circuit power consumption over all gates by using Vdd, Vth, W parameter scaling for SoCs. The most difficulties come from the non-linear interactions of the object parameters and their adaptation into the very large circuit. For example, each gate has at least four non-linear variables (Vdd, Vth, W, Delay) and after logic synthesis of target system, each functional module (i.e., ALU, Adder, Multiplier,…) might have several thousand number of gates. All path enumerations for the large size gate-level net list is NP-Hard. Therefore, this report demonstrates that the hierarchical approach with circuit partitioning and graph theory scheme can solve this problem within reasonable simulation time as shown in Fig. 4.

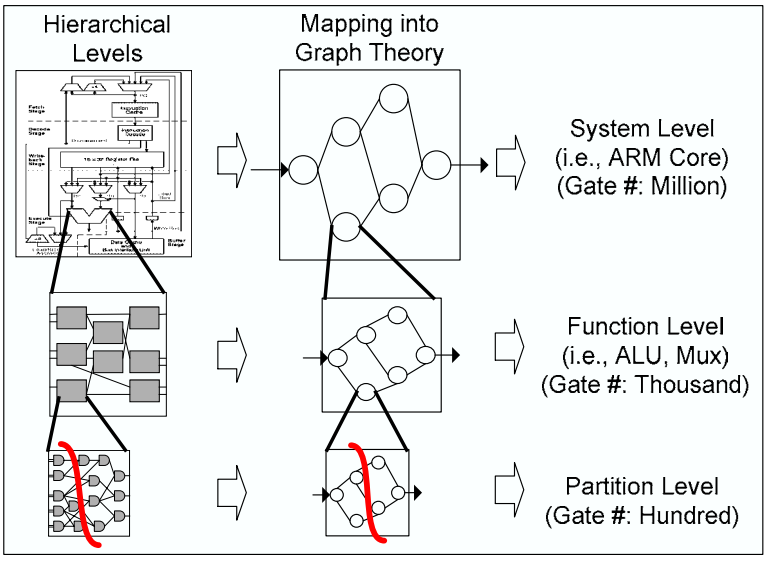


Figure 4. Hierarchical Approach

The problems are; 1) the delay assignment mechanism for each module and sub-module in the hierarchical environment and 2) the effective joint optimization of the Vdd, Vth, and W at the gate level. After optimal maximum delay assignment for each module, we try to reduce the slack time of each module as small as possible. The power saving and slack time tradeoff is presented in Fig. 5(a).

In this report, for the delay assignment scheme, we introduce the Transition Density Path (TDP) based maximum delay assignment algorithm for each module for the best power reduction because dynamic power consumption depends mostly on the switching activities. Almost all conventional circuit optimization approach is based on the critical delay [19] or fan-in/out counts [3] for the delay assignment. Fig. 4(b) shows critical time based delay assignment and Fig. 4(c) presents the TDP based delay assignment scheme.

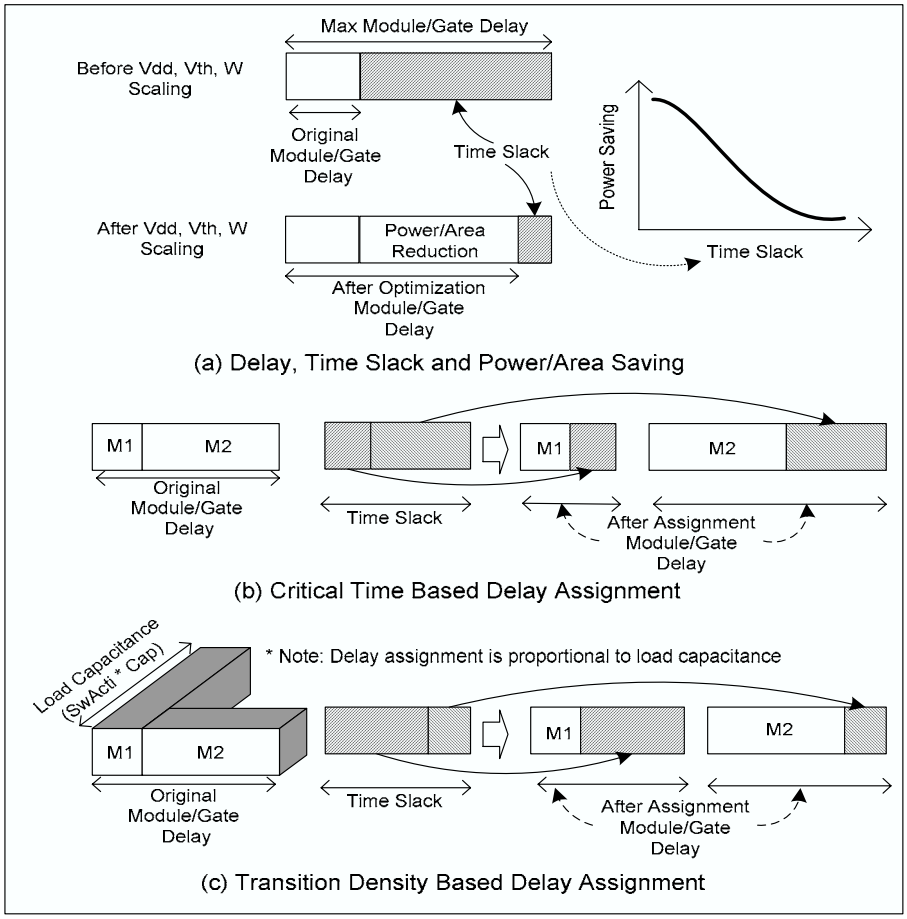


Figure 5. Maximum Delay Assignment for Ultra-low Power Optimization

The strategy for the gate-level optimization is to find iteratively, using binary search, the optimal combination of Vdd, Vth, and W for each gate that meets the maximum delay condition while achieving minimum total power (Static, Dynamic, and Short-circuit power) dissipation. Fig. 6 shows the rationale for the gate-level low-power optimizer.

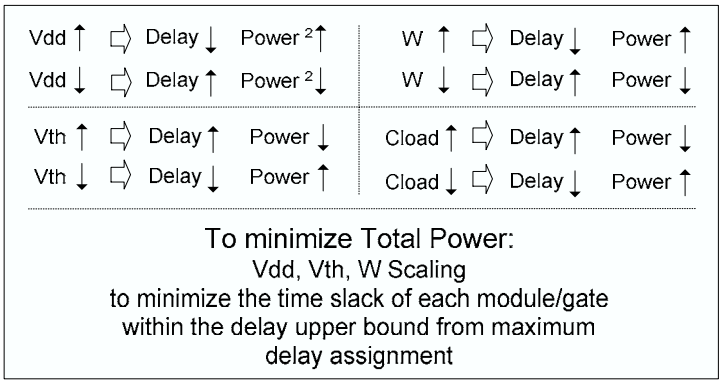


Figure 6. Gate-level Power Optimization Rationale

IV. Key Contribution

In this report, we focus on the following issues through our experimental demonstration:

1) Hierarchical low-power design methodology using technology scaling. This allows optimization of very large integrated circuit designs like SOCs.
2) Algorithm for module-level delay assignment to minimize total power.
3) Impact of the technology parameter optimization for the ultra-low power SoCs.
4) Impact of the interactions between logic synthesis and circuit/device level optimization for power aware system design
5) Impact of the Transition Density Path based optimization
6) Impact of the run-time saving for the simulation through hierarchical partitioning

V. Methodology and Algorithm

As shown in Fig. 1, we followed the overall optimization methodology procedure. In this report, we address more closely the transition density path generation method, the module-level delay assignment algorithm, and the gate level optimizer in the optimization procedure, which are key steps for the problem solving procedure.

1. System Specification and RTL (register transfer level) Design

For the System Specification and RTL Design, we used the cycle accurate Verilog model of the ARM-like RISC architecture, referring from [20,21,22]. The reason we have considered the ARM architecture is that ARM powered cores can be found at the heart of the industry's leading products from mobile phones to portable computing devices in the race to bring a new generation of wireless products to market. Currently ARM cores are being developed into more than 78% of cell phones worldwide. We synthesized the RTL core using Synopsys Design Compiler [23] targeted towards a 0.25-micron TSMC library from LEDA Systems [24]. After the logic synthesis, we extracted the gate level net list for each functional unit and then, interfaced the net lists to our gate level power optimization CAD tool properly.

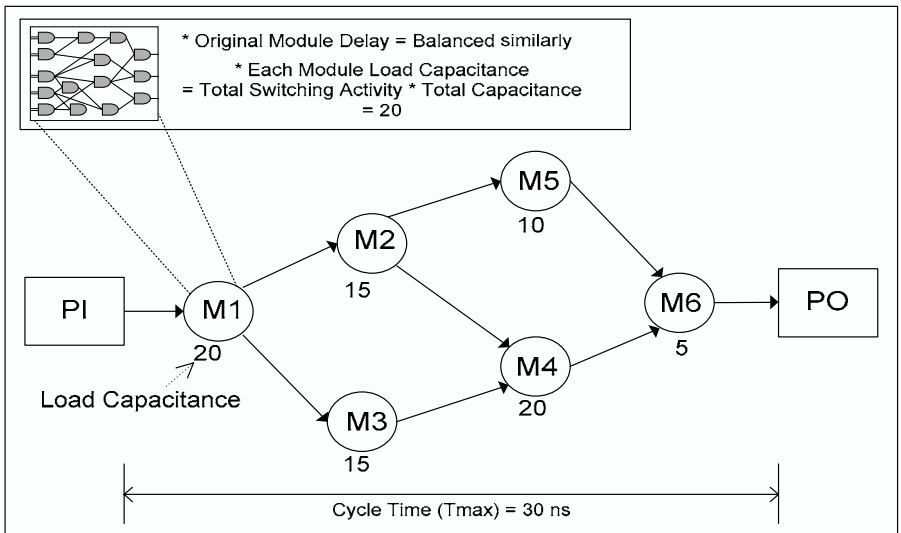
2. Transition Density Path Generation

A *Monte Carlo simulation* is conducted for the transition activity generation of each path list as described in [7]. This approach consists of applying randomly generated input patterns at the primary inputs of the circuit and monitoring the switching activity per time interval T using a simulator. Based on the assumption that the switching activity provided by the circuit over any period T has a normal distribution, and for a desired percentage error in the activity estimate and a given confidence level, the number of required simulation vectors is estimated. Our gate level activity profile simulator generate to calculate the activities at the internal nodes for each module of the circuit. Simulation based approach is accurate and

capable of handling various device models, different circuit design styles. After the activity profiling, Path enumeration is conducted at gate level.

3. Maximum Delay Assignment

Fig. 7 presents an example of the module level delay assignment algorithm. In the first step, each module is sorted by the amount of load capacitance of each module (step 1). According to the priority of each module, we assign maximum delay with the "objective function" and "delay assignment" formula in Fig. 7 (Step 2 and 3). Then we look at the local improvement by local search (step 4). If all modules' delays are assigned, conduct the technology parameter optimization at the gate level (step 5). Finally, we generate the power/area saving values and optimal parameters. In the algorithm, each module (M1,…,Mi) could be a functional module or its sub-partitioning, the total physical capacitance of a module can be the sum of the fan-in/out counts inside the module, and the load capacitance of each module can be calculated by multiplying the total switching activities by the total fan-in/out net counts.



Object : Assign max delay of each module for max power saving
(Note: slack time = power/area saving)

Object Function = $\frac{\alpha(M1)\times C(M1)}{D(M1)} = \frac{\alpha(M2)\times C(M2)}{D(M2)} = \cdots = \frac{\alpha(M6)\times C(M6)}{D(M6)}$

$$D(M1)+D(M2)+\cdots+D(M6)\le T_{\max}$$

(α = switching activity, C = capacitance, D = delay)

Delay Assignment =

$$\frac{\alpha(M_i)\times C(M_i)}{\text{Total Load Capacitance Sum in Path}}\times(T_{\max}\text{ - Assigned Delay Sum})$$

Step 1: Module Priority queue for each module by load capacitance

M4, M1 (20)
M2 (15)
M3 (10)
M5,M6 (5)

Step 2: - Select M4
- Path Priority queue for each path with M4 by load capacitance

Path1: M1,M2,M4,M6 (60)
Path2: M1,M3,M4,M6 (55)

- Select Path1
- Delay of M4 = (20/60)*30 = 10 ns

Step 3: - Repeat Step2 for all modules
- Delay of M1 = (20/40)*20 = 10 ns
- Delay of M2 = (15/20)*10 = 7.5 ns
- Delay of M3 = (10/15)*10 = 6.66 ns
- Delay of M5 = 10 ns
- Delay of M6 = 2.5 ns

Step 4: - Local search improvement
- Increase 6.66 to 7.5 for M3

Step 5: - Go to Gate level optimization (Vdd, Vth, W Scaling)
with this Max delay of each module

Figure 7. An Example of Module Level Delay Assignment

4. Gate-Level Power Optimization through CMOS Technology Scaling

After the maximum delays have been assigned to each gate in the circuit, we optimize each gate individually for minimum power. The strategy is to find iteratively, using binary search, the optimal combination of Vdd, Vth, and W for each gate that meets the maximum delay condition while achieving minimum power dissipation. This strategy is based on the observation that power consumption and delay are monotonic functions of Vdd, Vth, and W, individually, other parameters being fixed. Since it is impractical to have more than one power supply or threshold voltage in the circuit, we keep only one global value of Vdd and Vth. However, the algorithm could be easily modified to allow the use of multiple threshold values in the circuit if desired. The algorithm is outlined in Fig. 8.

```
V_dd Range ←— [0.6v-3.3v]
For M steps do
  V_dd ←— MID(V_dd Range)
  V_th Range ←— [0.1v-0.7v]
  For M steps do
    V_th ←— MID(V_th Range)
    For all gates do
      W Range ←— [1-100]
      For M steps do
        If delay(g) smaller than max delay(g), then
          W Range ←— LOWER(W Range)
        else
          W Range ←— HIGHER(W Range)
        endIf
      endFor
    endFor
    If All g: delay bounded and total energy decreased
    then
      V_th Range ←— HIGHER(V_th Range)
    else
      V_th Range ←— LOWER(V_th Range)
    endIf
  endFor
  If All g: delay bounded and total energy decreased
  then
    V_dd Range ←— LOWER(V_dd Range)
  else
    V_dd Range ←— HIGHER(V_dd Range)
  endIf
endFor
```

Figure 8. Technology Parameter Optimization Algorithm

In the procedure, *XRange* denotes the range of values that the variable *X* can take. The subroutine *MID (XRange)* returns the central value of *XRange*, while *LOWER (XRange)* and *HIGHER (XRange)* set *XRange* to its lower and higher subranges, splitting it at *MID (XRange)*. The algorithmic complexity of this procedure depends on the number of iteration steps that we allow for convergence to the optimal values. Assuming that Vdd, Vth and W are each constrained to 2^M quantized values, it takes $O(M^3)$ simulations of the entire circuit to obtain the final optimal values. This is many orders of magnitude lower than the complexity of any direct or random search algorithm that may be used to search for the optimal solution. For evaluation purposes, we have also implemented an optimization tool for the above problem using multiple-pass simulated annealing. Our approach performed better than annealing over almost all circuits.

VI. Experimental Results and Discussion

We used several tools for the RTL description, the functional verification and the logic synthesis and developed some interface programs and simulators for the proposed hierarchical optimization with C and C++/STL on Ultra-80 Unix machine. Some ARM based arithmetic modules are used for the benchmark circuits. For the range of the technology parameter values, we refer to the 2000 updated version of ITRS (International Technology Roadmap for Semiconductors). We used Verilog for the RTL design, VCS (Synopsys) for the verilog functional simulation, and design analyzer (Synopsys) with 0.25 micron TSMC library for the logic synthesis.

Table I demonstrates the efficiency and effectiveness of the technology parameters optimization with the proposed design flow. Table I (a) shows the static, dynamic and short-circuit power consumption of the circuits under minimum total power for two different activity levels at the circuit inputs, assuming a fixed

threshold voltage of 700mV. The power consumption metrics were obtained by optimizing the device widths and supply voltage to minimize power while meeting a cycle time constraint obtained from the logic synthesis. It is assumed that the activity levels are the same over all the inputs to the circuit. The activities at the internal nodes were calculated as described in Section V. Table I (a) was generated to serve as a basis for comparative evaluation of our power minimization algorithm. In Table I, the mean of the device widths (w) and the standard deviation (σ_w) were taken over all the devices that compose the circuit.

Table I (b) shows the static, dynamic and short-circuit power components yielded by our algorithm for all the benchmark logic networks of Table I (a). It is seen that the total energy dissipation of the circuits reduces by factors larger than 20-70x compared to the situation where only gate sizing was used to decrease the power consumption without any performance loss. Table I (b) also indicates that the total area of the circuit is lesser. (The area was estimated to be proportional to the sum of all the device widths of the gates in the circuit.) It can also be seen that the static and the dynamic power components are approximately equal, thus confirming the previously discussed physics of the optimization process (Section V). We also note that the power savings are more for higher input activity levels. For reasons of practical utility, the data in Table I (b) was obtained assuming the use of a single power supply and threshold voltage across all the gates.

Table II shows the influence of the logic level design on the optimization. We synthesized 16-bit lookahead adder with different critical delay constraints and then, optimized through our hierarchical power optimization tool. Table III shows the effectiveness of the proposed module-level TDP-based delay assignment scheme. Approximately 36%-39% more power reduction than conventional optimization approaches could be obtained.

TABLE I
Impact of Technology Parameters Optimization

(a) Before Optimization without Sub-partitioning (Fixed Vdd:3.3v, Vth:0.7v)

System Module	Gates/ Depth	Delay (ns)	Input Activity	ϖ, σ_ω	Power Dissipation			
					Static	Dynamic	Short-ckt	Total
4 - Full Adder	106/48	3.36	0.5	17.9, 24.5	2.09x10E-20	4.37x10E-11	2.15x10E-12	4.59x10E-11
			0.05	17.9, 24.5	2.09x10E-20	4.33x10E-12	2.13x10E-13	4.54x10E-12
16- Full Adder	1030/ 93	6.98	0.5	7.4, 5.6	8.60x10E-20	9.80x10E-11	8.99x10E-11	1.87x10E-10
			0.05	7.4, 5.6	8.60x10E-20	9.56x10E-11	7.51x10E-12	1.03x10E-11
16 - Look ahead	1838/ 81	7.0	0.5	5.9, 6.2	1.48x10E-19	7.65x10E-10	9.33x10E-11	8.58x10E-10
			0.05	5.9, 6.2	1.48x10E-19	1.39x10E-10	9.29x10E-12	1.48x10E-10
64 - ALU	3417/ 426	28.6	0.5	N/A	N/A	N/A	N/A	N/A
			0.05	N/A	N/A	N/A	N/A	N/A

(b) After Optimization with Sub-partitioning (Vdd:0.6-3.3v, Vth:0.1-0.7v)

System Module	Delay (ns)	Number of Sub-partitioning	Input Activity	Vdd,Vth	ϖ, σ_ω	Total Power	Reductions		
							Power	Area	Run-Time
4 - Full Adder	3.01	1	0.5	0.625, 0.1	6.61, 8.14	6.52x10E-13	70.4x	63.2%	0%
			0.05	0.625, 0.1	6.61, 8.14	7.13x10E-14	63.7x	63.2%	0%
16- Full Adder	7.14	4	0.5	0.625, 0.1	5.98, 5.03	8.70x10E-13	21.5x	19.1%	55.8%
			0.05	0.6, 0.1	5.02, 4.03	6.30x10E-13	16.3x	32.1%	57.3%
16 - Look ahead	5.49	4	0.5	0.65, 0.1	5.21, 5.70	1.80x10E-12	47.6x	11.7%	63.1%
			0.05	0.625, 0.12	3.16, 2.05	1.78x10E-12	39.1x	46.4%	72.5%
64 - ALU	20.07	8	0.5	0.625, 0.1	6.09, 9.04	9.91x10E-09	Infinite	Infinite	Infinite
			0.05	0.625, 0.1	4.91, 6.14	6.47x10E-10	Infinite	Infinite	Infinite

TABLE II
Impact of Logic Level Design during Circuit/Device Level Optimization [Using Different Gate-Net-Lists from Synthesis Designs with different delay constraints]

(a) Before Optimization (Fixed Vdd:3.3v, Vth:0.7v)

System Module	Gates/ Depth	Delay (ns)	Input Activity	ϖ, σ_ω	Power Dissipation			
					Static	Dynamic	Short-ckt	Total
16 - Look ahead (1)	1838/ 81	7.0	0.5	5.9, 6.2	1.48x10E-19	7.65x10E-10	9.33x10E-11	8.58x10E-10
			0.05	5.9, 6.2	1.48x10E-19	1.39x10E-10	9.29x10E-12	1.48x10E-10
16 - Look ahead (2)	1871/ 75	6.3	0.5	7.2, 4.9	1.88x10E-19	8.15x10E-10	9.63x10E-11	9.11x10E-10
			0.05	7.2, 4.9	1.88x10E-19	5.39x10E-10	9.59x10E-12	5.49x10E-10
16 - Look ahead (3)	1928/ 69	5.9	0.5	7.9, 5.3	1.63x10E-19	8.59x10E-10	9.91x10E-11	9.58x10E-10
			0.05	7.9, 5.3	1.63x10E-19	5.50x10E-10	9.90x10E-12	5.59x10E-10

(b) After Optimization (Vdd:0.6-3.3v, Vth:0.1-0.7v)

System Module	Gates/ Depth	Delay (ns)	Input Activity	Vdd, Vth	ϖ, σ_ω	Total Power	Power Reduction	Area Reduction
16 - Look ahead (1)	1838/ 81	7.0	0.5	0.65, 0.1	5.21, 5.70	1.80x10E-12	47.6x	11.7%
			0.05	0.625, 0.12	3.16, 2.05	3.78x10E-12	39.1x	46.4%
16 - Look ahead (2)	1871/ 75	6.3	0.5	0.625, 0.1	7.1, 6.90	5.98x10E-11	15.2x	1.3%
			0.05	0.625, 0.1	6.16, 6.05	3.08x10E-11	17.8x	14.4%
16 - Look ahead (3)	1928/ 69	5.9	0.5	0.625, 0.1	7.81, 5.01	6.19x10E-11	14.7x	1.1%
			0.05	0.625, 0.1	3.16, 2.05	7.78x10E-11	7.0x	60.0%

TABLE III
Impact of Transition Density Based Optimization

(a) Critical Delay Based Optimization (Vdd:0.6-3.3v, Vth:0.1-0.7v)

System Module	Delay (ns)	Gates/ Depth	Input Activity	Vdd, Vth	ϖ, σ_ω	Total Power
64 - ALU	20.07	3417/ 426	0.5	0.625, 0.1	5.89, 4.94	1.56x10E-08
			0.05	0.625, 0.1	4.81, 3.88	1.07x10E-09

(b) Fan-in/out Based Optimization (Vdd:0.6-3.3v, Vth:0.1-0.7v)

System Module	Delay (ns)	Gates/ Depth	Input Activity	Vdd, Vth	ϖ, σ_ω	Total Power
64 - ALU	20.07	3417/ 426	0.5	0.625, 0.1	6.19, 5.04	9.98x10E-09
			0.05	0.625, 0.1	4.98, 4.74	8.02x10E-10

(c) Transition Density Based Optimization (Vdd:0.6-3.3v, Vth:0.1-0.7v)

System Module	Delay (ns)	Gates/ Depth	Input Activity	Vdd, Vth	ϖ, σ_ω	Total Power
64 - ALU	20.07	3417/ 426	0.5	0.625, 0.1	6.09, 9.04	9.91x10E-09
			0.05	0.625, 0.1	4.91, 6.14	6.47x10E-10

*Note: Scheme (c) is better than scheme (a)
around 36% - 39% in total power reduction

VII. Conclusion and Future Work

This report presents an efficient hierarchical low-power design flow and a novel transition density based ultra-low power optimization algorithm through CMOS technology parameter scaling for SOCs. Experimental results show that the algorithm yields reduction in power by a factor from 20x to 70x and presents run-time saving around 50% or more across few functional sub-modules. Consequently the new power minimization technique provides the following advantages: 1) power reduction is achieved without performance loss, 2) both static and dynamic components are optimized, 3) the algorithm is fast. Moreover, the energy savings are in addition to what is achievable by activity minimization through assorted architectural and algorithmic techniques. Future work will include application-specific and architecture- driven issues with this technology parameter scaling techniques.

References

[1] K.Imai, K. Yamaguchi, T, Kudo, et al., "CMOS device optimization for system-on-a-chip applications," IEDM Technical Digest International, 2000, pp. 455 -458.

[2] T. Nishimura, "Trend of the CMOS process technology for system on a chip," Conference on Ion Implantation Technology, 2000, pp. 20-24.

[3] P. Pant, V. De, and A. Chatterjee, "Simultaneous power Supply, threshold voltage, and transistor size optimization for low-power operation of CMOS circuits," *IEEE Trans. On VLSI Systems*, vol. 6, no. 4, pp. 538-545, December 1998.

[4] A. Salek, J. Lou, M. Pedram, "An integrated logical and physical design flow for deep submicron circuits," *IEEE Trans. On CAD of Integrated Circuits and Systems*, vol. 18, no. 9, pp. 1305-1315, September 1999.

[5] Rao and F. Kurdahi, "Hierarchical design space exploration for a class of digital systems," *IEEE Trans. On VLSI Systems*, vol. 1, no. 3, pp. 282-295, September 1993.

[6] J. Burr and J. Shott, "A 200 mv self-testing encoder-decoder circuit using Stanford ultra low power CMOS,," in Proc. ISSCC: Tech. Dig., Feb. 1994, pp. 84-85.

[7] J.M. Rabaey and M. Pedram, *Low Power Design Methodologies*, Kluwer Academic Publishers, 1996, pp 21-64,130-160.

[8] Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 473-484, April 1992.

[9] Raghunathan, N.K. Jha, and S. Dey, *High-Level Power Analysis and Optimization,* Kluwer Academic Publishers, 1998, pp 1-25.

[10] K. Roy and S.C. Prasad, *Low-Power CMOS VLSI Circuit Design,* John wiley & Sons, Inc., 2000, pp 201-252.

[11] T. Kobayashi and T. Sakurai, "Self adjusting threshold voltage scheme (SATS) for low voltage high speed operation," *IEEE CICC*, 1994, pp.271-.

[12] S. Mutoh, "1-V Power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, pp. 847-, April 1992.

[13] A. Fariborz, "A dynamic threshold voltage MOSFET (DTMOS) for ultra-low voltage operation," IEDM Tech., 1994, pp809-.

[14] L. Wei, Z. Chen, and K.Roy, "Double gate dynamic threshold voltage (DGDT) SOI MOSFETs for low power high performance designs," IEEE SOI conference, 1997, pp. 82-83.

[15] G. Yeap, *Practical Low Power Digital design*, Kluwer Academic Publishers, 1998, pp 85-116.

[16] S.S. Sapatnekar, V.B. Rao, P.M. Vaidya, and S Kang, "An exact solution to the transistor sizing problem ofr CMOS circuits using convex optimization," *IEEE Trans. On CAD of Integrated Circuits and Systems*, vol. 12, no. 11, pp. 1621-1634, September 1993.

[17] A. Bhavnagarwala, V. De, B. Austin, and J. Meindl, "Circuit techniques for CMOS low power GSI," in Proc. Int. Symp. Low Power Electron. Design: Dig. Tech. Papers, Aug. 1996, pp. 193-196.

[18] N. Hendenstierna and K.O. Jeppson, "CMOS circuit speed and buffer optimization," IEEE Trans. Computer-Aided Design, vol. 6, pp. 270-281, Mar. 1987.

[19] L.Wei, Z. Chen, K Roy, M.C. Johnson, Y. Ye, and V.K. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *IEEE Trans. On VLSI Systems*, vol. 7, no. 1, pp. 16-24, March 1999.

[20] ARM Ltd., http://www.arm.com

[21] http://www.eecs.umich.edu/~jringenb/power/

[22] Chang. N, Kim. K, Lee H. G., "Cycle Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI", *Proceedings of the International Symposium on Low Power Electronics and Design,* pp 185-190, July 2000.

[23] Synopsys, Inc., http://www.synopsys.com

[24] LEDA Systems Inc., http://www.ledasys.com

[25] T. Sakurai and A.R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," IEEE J. Solid-State Circuits, vol.25, pp. 584-594, Apr. 1990.

Appendix

A. Energy Model Equations

The equations used to compute the dynamic and static energy dissipations of a gate are described next. Similar models have been presented and analyzed in a recent work by [17]. It is assumed that the gates are simple multi-input gates with symmetric series or parallel pull-up and pull-down MOSFET configurations. Contributions of subthreshold leakage through the MOSFET channel as well as the leakage across the device drain junctions to static dissipation are included.

1) Static Dissipation of Gate G_i ($i \in N$):

$$E_{S_i} = V_{dd} W_i I_{off} / f_c$$

- V_{dd} is the power supply voltage;
- $w_i \geq 1$ is the device width (adjusting w_i scales the widths of all the transistors in G_i);
- I_{off} is the off current per unit width;
- f_c is the clock frequency.

2) Dynamic and Short-Circuit Dissipation of Gate G_i ($i \in N$)

$$E_{d_i} = \frac{1}{2} a_i V_{dd}^2 (1 + K_{SC}) \cdot \left[w_i \{ C_{PD_i} + (f_{ii} - 1) C_{m_i} \} \sum_{j=1}^{f_{oi}} (w_{ij} C_{t_{ij}} + C_{INT_{ij}}) \right]$$

- α_i is the activity factor of the gate output;
- f_{ii} and f_{oi} are the number of fanins and fanouts;

- $w_{ij} \geq 1$ is the device width the gate at the *j*th fan-out;
- C_{DPi} is the sum of the overlap, junction and finging capacitance at the output node per unit width;
- C_{mi} is the intermediate node capacitance of series connected MODFET's in multiple fan-in gates;
- C_{tii} is the input capacitance per unit width of the gate being driven by the *j*th fan-out;
- C_{INTii} is the interconnect capacitance at the *j*th fan-out;
- K_{SC} is the coefficient for short-circuit dissipation [18].

B. Delay Model Equations

We use a transregional model for estimating the worst-case signal propagation delay through a gate. The delay model has been derived using an extension of the alpha-power law saturation drain current model [25] to the subthreshold region. The drain current model incorporates effects of high-field and quasi-ballistic (velocity overshoot) carrier transport in the MOSFET channel. All components of the delay, namely, 1) the delay due to switching MOSFETs, 2) the distributed interconnect RC delay, 3) the time of flight delay, 4) the delay component due to the non-zero rise time of the input signal are considered.

$$t_{d_i} = \left[\frac{1}{2} - \frac{1-\frac{V_{TS_i}}{V_{dd}}}{1+\alpha}\right] \max_{j\in(1,f_{ii})}\{t_{d_{i,j}}\} + \frac{V_{dd}/2}{I_{Diw} - f_{ii}\beta I_{off}} \cdot \left[C_{DP_i} + \frac{1}{w_i}\sum_{j=1}^{f_{oi}}(w_{ij}C_{t_{ij}} + C_{INT_{ij}})\right]$$

$$+ \max_{j\in(1,f_{oi})}\{t_{d_{i,j}}\}\left[R_{INT_{ij}}(w_{ij}C_{t_{ij}} + \frac{1}{2}C_{INT_{ij}}) + \frac{L_{INT_{ij}}}{v}\right] + \frac{1}{2}C_{m_i}V_{dd}\sum_{j=1}^{f_{ii}-1}\frac{1}{I_{Diw}(j)}$$

- t_{dij} is the delay of the gate at the *j*th fan-in;
- $1 \leq \alpha \leq 2$ is the velocity saturation coefficient;

- $\beta \geq 1$ is the pMOSto nMOS width ratio;
- $I_{Diw}(f_{ii})$ is the switching drain current per unit width;
- L_{INTij} is the interconnection length at the jth fan-out;
- R_{INTij} is the interconnection resistance at the jth fan-out;
- v_{tii} is the propagation velocity through the interconnect;
- V_{TSi} is the threshold voltage of the ith gate.